



TO52 - PROJET OPERATIONNEL

FISE - INFO

Thème :

CONCEPTION D'UNE API WEB EN PYTHON

DOSSIER D'ANALYSE

Etudiants

AMOUZOUVI Kodjo Franc

KOMISSA ZOTSU Essi

Responsable du sujet

M. Didier ASSOSSOU

SOMMAIRE

I.	INTRODUCTION	4
II.	CONTEXTE DU PROJET.....	4
	Objectifs du projet	5
III.	ANALYSE ET CONCEPTION.....	5
III.1.	Diagramme de cas d'utilisation	5
III.1.1.	Identification des acteurs	5
III.1.2.	Diagramme de cas d'utilisation du système.....	6
III.1.3.	Description des cas d'utilisation	7
III.1.3.1.	Cas d'utilisation : "S'authentifier"	7
III.1.3.2.	Cas d'utilisation : "Créer un évènement"	8
III.1.3.3.	Cas d'utilisation : "Payer un ticket"	9
III.2.	Diagramme de séquences	11
III.2.1.	Diagramme de séquence du cas : "S'authentifier"	12
III.2.2.	Diagramme de séquences du cas : "Créer un évènement"	13
III.2.3.	Diagramme de séquences du cas : Payer un ticket	14
III.3.	Diagramme de classes	15
III.3.1.	Identification des classes.....	15
III.3.2.	Détails des classes.....	16
III.4.	Diagramme d'activités	18
III.4.1.	Diagramme d'activités du cas : "S'authentifier"	19
III.4.2.	Diagramme d'activités du cas : "Créer un évènement"	20

III.4.3. Diagramme d'activités du cas : "Payer un ticket"	21
III.5. Diagramme de déploiement.....	22

LISTE DES FIGURES

Figure 1 : Diagramme de cas d'utilisation du système	6
Figure 2 : Diagramme de séquences du cas : S'authentifier	12
Figure 3 : Diagramme de séquences du cas : Créer un évènement.....	13
Figure 4 : Diagramme de séquences du cas : Payer un ticket.....	14
Figure 5 : Diagramme de classes	15
Figure 6 : Diagramme d'activités du cas : S'authentifier	19
Figure 7 : Diagramme d'activités du cas : Ajouter un évènement	20
Figure 8 : Diagramme d'activités du cas : Payer un ticket	21
Figure 9 : Diagramme de déploiement.....	22

I. INTRODUCTION

Le présent document est un document d'analyse portant sur la conception d'une API Web qui a pour but de fournir des informations sur des événements. Cette analyse a pour but d'identifier les objectifs de notre API, les différentes fonctionnalités qu'il doit offrir, ainsi que les contraintes techniques et fonctionnelles auxquelles notre système doit se conformer. Le document est structuré en plusieurs sections : la première présente une description générale du sujet ; la deuxième décrit les méthodes d'analyse et les approches utilisées ; la troisième détaille les utilisateurs du système ainsi les fonctionnalités que le système doit offrir pour répondre aux besoins des utilisateurs et la dernière présente les contraintes techniques et fonctionnelles auxquelles le système doit se conformer.

L'analyse présentée dans ce document constitue une étape clé dans le processus de développement du système et fournira des orientations importantes pour la conception, la mise en œuvre et les tests du système. Nous espérons que cette analyse sera utile pour les parties prenantes impliquées dans le projet et contribuera à la réussite du système.

II. CONTEXTE DU PROJET

Notre sujet s'intitule : « **CONCEPTION D'UNE API WEB EN PYTHON** ». Elle permettra à partir d'une plateforme Web de fournir des services à l'utilisateur en utilisant les données issues de requêtes vers une base de données.

La plateforme Web avec laquelle communiquera notre API est une plateforme Web d'organisation événementielle dont le but ultime est de fournir des informations sur des événements. Grâce à notre API, la plateforme Web sera en mesure de fournir également des services tels que la consultation des événements, l'achat de tickets pour les événements, la newsletter, la publicité, etc. Nos données proviennent d'une base de données MySQL qui contient toutes les informations nécessaires à notre étude.

Objectifs du projet

L'objectif globale de la solution est de permettre aux utilisateurs d'avoir des informations sur les évènements.

Plus spécifiquement :

- Afficher les évènements à venir
- Permettre aux utilisateurs d'acheter des tickets

III. ANALYSE ET CONCEPTION

L'objectif dans cette section est de réaliser les différents modèles UML afin de schématiser le fonctionnement de notre solution.

III.1. Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation traduit tout ce que l'utilisateur exprime comme action sur le logiciel ou le système à modéliser. C'est une représentation faisant intervenir les acteurs et les cas d'utilisation. Il traduit les besoins des utilisateurs vis-à-vis du système développé. Le diagramme de cas d'utilisation fait intervenir la notion d'acteurs, de cas d'utilisation et de relation.

III.1.1. Identification des acteurs

Un acteur est une personne ou un système qui interagit avec le système en échangeant des informations en entrée comme en sortie. Pour notre projet nous avons recensé comme acteurs :

- L'administrateur de la plateforme
- L'utilisateur

III.1.2. Diagramme de cas d'utilisation du système

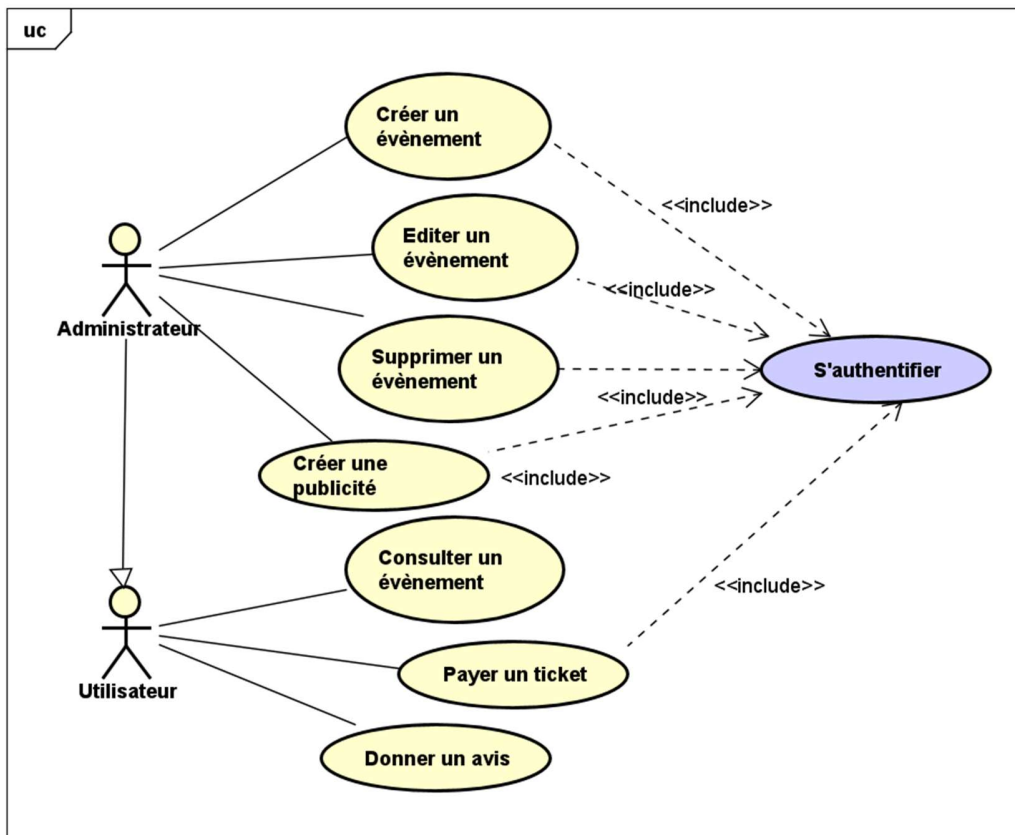


Figure 1 : Diagramme de cas d'utilisation du système

Notre diagramme de cas d'utilisation contient six cas d'utilisations qui sont répartis entre les deux principaux acteurs. Le cas d'utilisation « S'authentifier » est réalisé par l'administrateur avant la réalisation de tous ses autres cas d'utilisation. L'utilisateur n'effectuera le cas « S'authentifier » que s'il a besoin d'effectuer un achat. Par contre, il pourra effectuer le cas « Consulter un évènement » sans être connecté.

L'administrateur pourra effectuer les cas d'utilisation d'un simple utilisateur en plus de ses propres cas d'utilisation d'où la relation d'héritage entre les deux acteurs.

III.1.3. Description des cas d'utilisation

La description textuelle des cas d'utilisation consiste à expliquer de manière littérale les cas d'utilisation de notre projet afin de les détailler et en faciliter leur compréhension. Les différents cas d'utilisation de notre système sont :

- S'authentifier
- Créer un évènement
- Consulter un évènement
- Modifier un évènement
- Supprimer un évènement
- Payer un ticket
- Créer une newsletter
- Créer une publicité
- Donner un avis

III.1.3.1. Cas d'utilisation : "S'authentifier"

Ce cas d'utilisation permet d'accéder à la page d'accueil principale de notre site web. Il est effectué par les deux acteurs. Deux préconditions sont nécessaires à la réalisation de ce cas d'utilisation :

- Le site web est fonctionnel
- Les acteurs sont enregistrés dans la base de données

Scénario nominal

1. L'acteur demande à se connecter et le système affiche la page de connexion.

2. L'acteur saisit son adresse mail et son mot de passe puis appuie sur le bouton de connexion.
3. Le système vérifie si l'adresse mail au bon format. (A1)
4. Le système recherche l'adresse mail et le mot de passe dans la base de données. (A2)
5. Le système vérifie les droits d'accès.
6. La page d'accueil s'affiche à l'utilisateur/l'administrateur.

Scénario alternatif

A1 : l'adresse mail n'est pas conforme au format d'adresse mail.

1. L'adresse mail n'est pas conforme au format d'adresse mail.
2. Le système affiche le message "Adresse mail incorrecte".
3. Le déroulement reprend au point 2 du scénario nominal.

A2 : Identifiants incorrects.

1. Le système affiche le message « *l'adresse mail et/ou le mot de passe ne figure pas dans la base de données* ».

A la fin du cas d'utilisation une postcondition doit être vérifiée :

- L'acteur est redirigé vers la page requise.

III.1.3.2. Cas d'utilisation : "Créer un évènement"

Ce cas d'utilisation permet d'ajouter un nouvel évènement à la base de données. Il est effectué par l'administrateur. Les préconditions nécessaires à sa réalisation sont :

- La page d'enregistrement s'affiche à l'administrateur
- La base de données est fonctionnelle
- La liaison entre l'interface et la base de données est fonctionnelle

Scénario nominal

1. L'administrateur demande à créer un événement et le système affiche une page de création.
2. L'administrateur saisit les informations relatives à l'évènement puis appuie sur le bouton de validation.
3. Le système vérifie que tous les champs obligatoires sont remplis. (A1)
4. Le système enregistre les données dans la base de données.
5. Le nouvel événement s'affiche dans la liste des événements.

Scénario alternatif

A1 : Tous les champs obligatoires ne sont pas remplis

1. Un ou plusieurs champs obligatoires n'ont pas été remplis.
2. Le système affiche un message d'erreur « *Veillez renseigner ce champ* ».
3. Le scénario reprend au point 2 du scénario nominal.

A la fin du cas d'utilisation une postcondition doit être vérifiée :

- L'évènement est ajouté dans la base de données avec succès.

III.1.3.3. Cas d'utilisation : "Payer un ticket"

Ce cas d'utilisation permet à l'acteur de payer un ticket. Il est donc réalisable par les deux acteurs de notre système. Pour la réalisation de ce cas d'utilisation, plusieurs préconditions sont nécessaires.

- La page d'achat s'affiche à l'utilisateur
- Le ticket existe dans la base de données
- Le système de paiement est fonctionnel
- La liaison entre l'interface et la base de données est fonctionnelle

Scénario nominal

1. L'acteur clique sur le bouton « *Payer* » et accède à la page de paiement.
2. L'acteur remplit les informations d'achat et valide.

3. Le système vérifie les informations saisies. (A1)
4. Le système calcule automatiquement le montant à payer.
5. L'acteur choisit un mode de paiement.
6. L'acteur remplit les informations de paiement.
7. Le système vérifie les informations de paiement. (A2)
8. L'acteur valide la transaction. (A3)
9. Le système lui envoie un mail contenant les détails de l'achat et le ticket.

Scénario alternatif

A1 : Le nombre de tickets choisi dépasse la quantité en stock

1. L'utilisateur a choisi un nombre de tickets excédant la quantité en stock
2. Le système lui envoie un message d'erreur
3. Le scénario reprend au point 2 du scénario nominal

A2 : Les informations de paiement sont erronées

1. L'utilisateur a renseigné des informations de paiement erronées
2. Le système envoie un message d'erreur
3. Le scénario reprend au point 6 du scénario nominal

A3 : Le solde est insuffisant

1. L'utilisateur ne dispose pas d'un montant suffisant pour effectuer le paiement.
2. Le système renvoie un message d'erreur
3. Le scénario reprend au point 2 du scénario nominal

A la fin du cas d'utilisation une postcondition doit être vérifiée :

- L'acteur a effectué le paiement de ticket avec succès.

III.2. Diagramme de séquences

Le diagramme de séquence est un diagramme d'interaction dynamique. Il présente une description graphique d'un cas d'utilisation ou groupe de cas d'utilisation, en mettant l'accent sur la dynamique temporelle.

Dans un souci de simplification, on représente l'acteur principal à gauche du diagramme, et les acteurs secondaires éventuels à droite du système ; le but étant de décrire comment se déroulent les actions entre les acteurs ou objets.

III.2.1. Diagramme de séquence du cas : "S'authentifier"

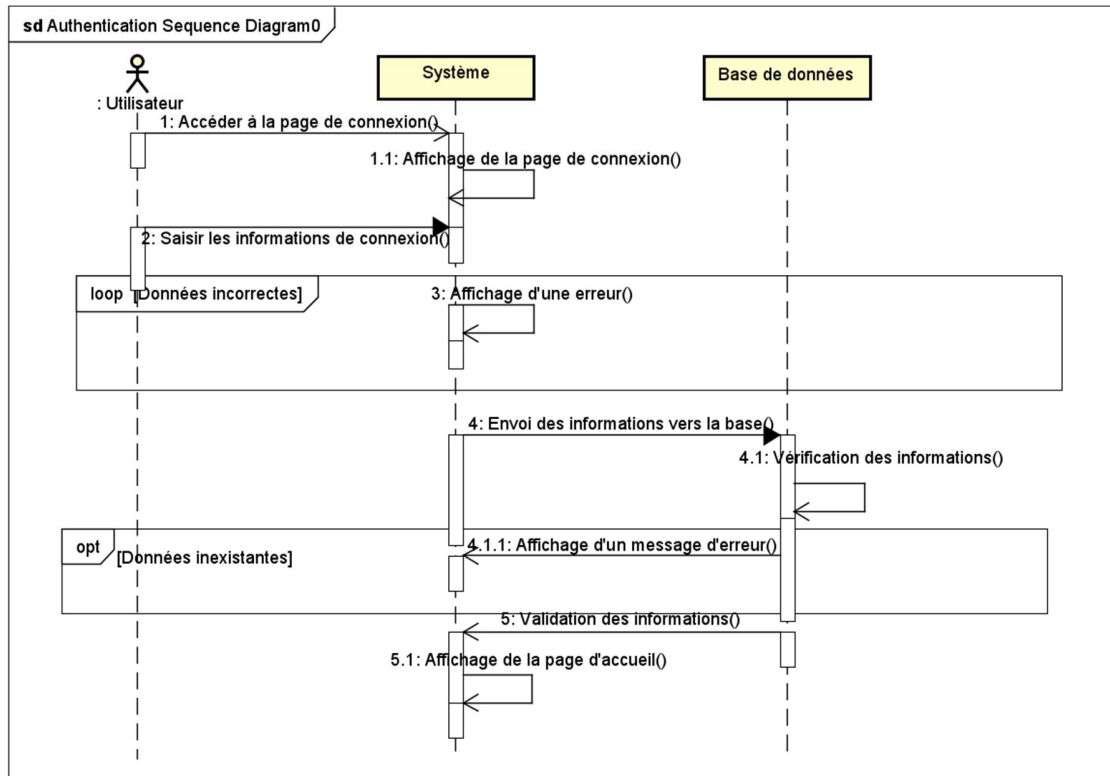


Figure 2 : Diagramme de séquences du cas : S'authentifier

Le diagramme de séquences du cas « S'authentifier » se réalise entre les acteurs, le système et la base de données. L'acteur accède à la page de connexion et saisit les informations de connexion. Le système vérifie ensuite le format de l'adresse mail et affiche une erreur si le format de l'adresse mail renseignée est incorrect. Au cas où il n'y a aucune erreur, le système envoie les informations de connexion à la base de données. La base de données vérifie si ces informations existent. Dans le cas où elles n'existent pas un message d'erreur est affiché à l'acteur. Dans le cas contraire, l'authentification est validée et la page requise s'affiche.

III.2.2. Diagramme de séquences du cas : “Créer un évènement”

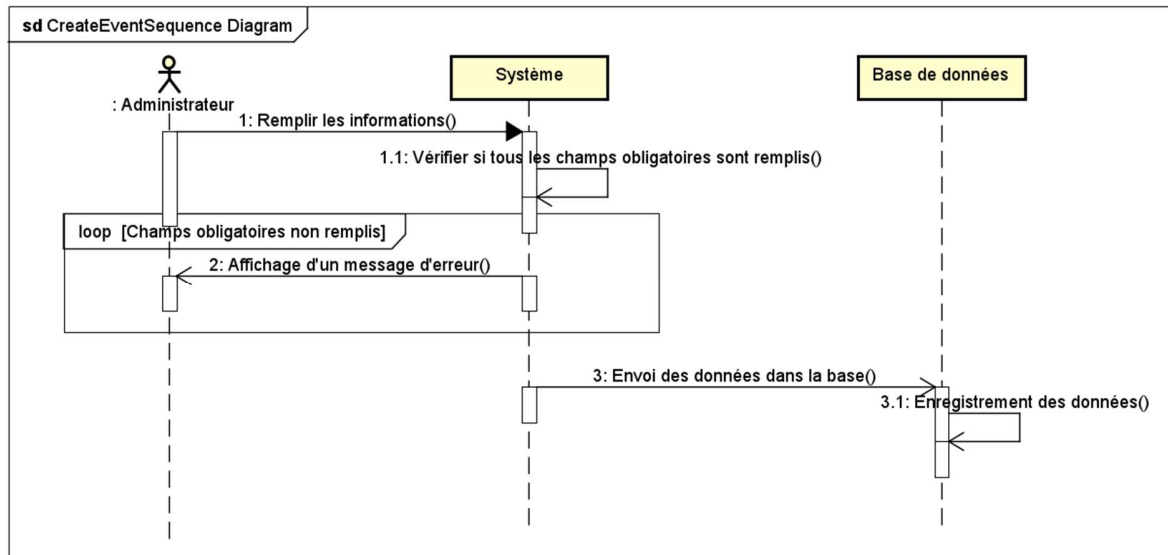


Figure 3 : Diagramme de séquences du cas : Créer un évènement

Le diagramme de séquences du cas « Créer un évènement » se réalise entre l'administrateur, le système et la base de données. L'administrateur renseigne les informations concernant le nouvel évènement et enregistre puis le système vérifie si tous les champs requis sont renseignés. Si un champ requis n'est pas renseigné, le système affiche un message d'erreur. Dans le cas contraire, les informations sont envoyées et enregistrées dans la base de données.

III.2.3. Diagramme de séquences du cas : Payer un ticket

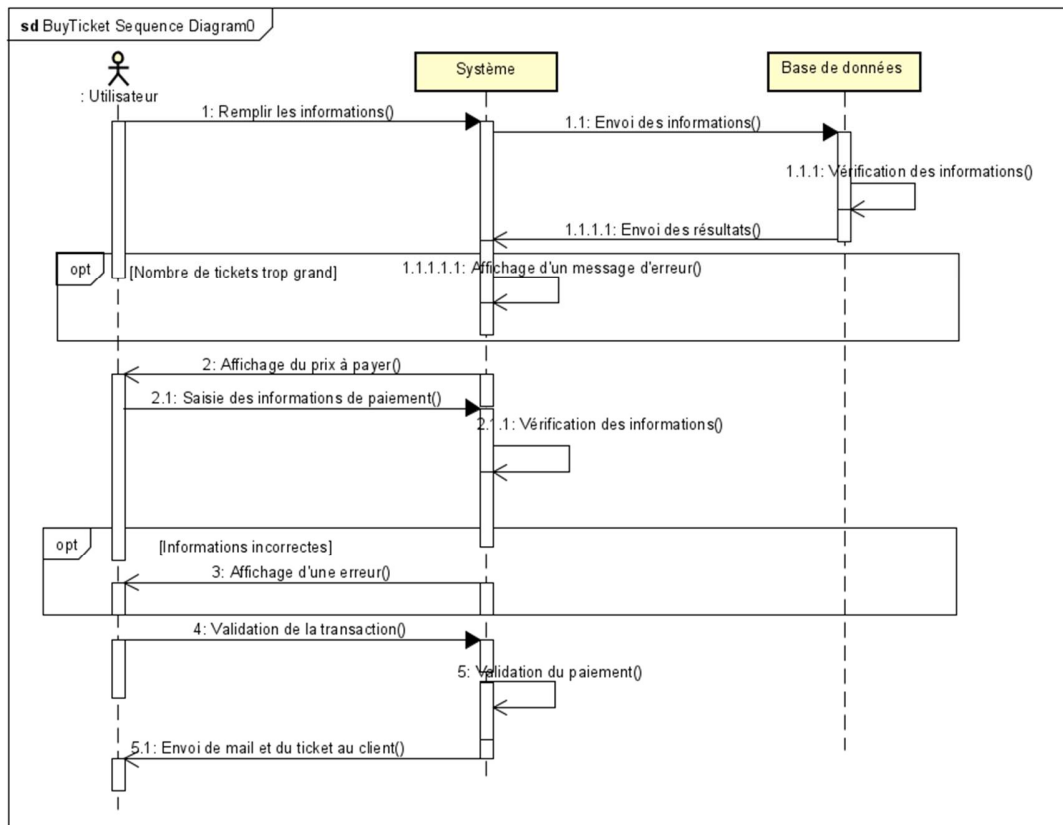


Figure 4 : Diagramme de séquences du cas : Payer un ticket

Le diagramme de séquences du cas « Payer un ticket » se réalise entre l'acteur, le système et la base de données. L'acteur accède à la page d'achat et remplit les informations et le système envoie les informations à la base de données. La base de données vérifie les informations telles que la quantité de tickets en stock, le solde de l'utilisateur. En cas d'erreur, un message est affiché à l'utilisateur qui corrige les informations renseignées. Dans le cas où toutes les informations sont correctes, l'acteur valide la transaction et le système lui envoie par mail la facture et le(s) ticket(s) acheté(s).

III.3. Diagramme de classes

Le diagramme de classes est un diagramme utilisé en UML pour représenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci. Ce diagramme fait partie de la partie statique de l'UML car il fait abstraction des aspects temporels et dynamiques. Une classe décrit les responsabilités, le comportement et le type d'un ensemble d'objets. Les éléments de cet ensemble sont les instances de la classe.

III.3.1. Identification des classes

Nous avons huit principales classes : Evènement, Type, Ticket, Utilisateur, Administrateur, Paiement, Avis, Publicité.

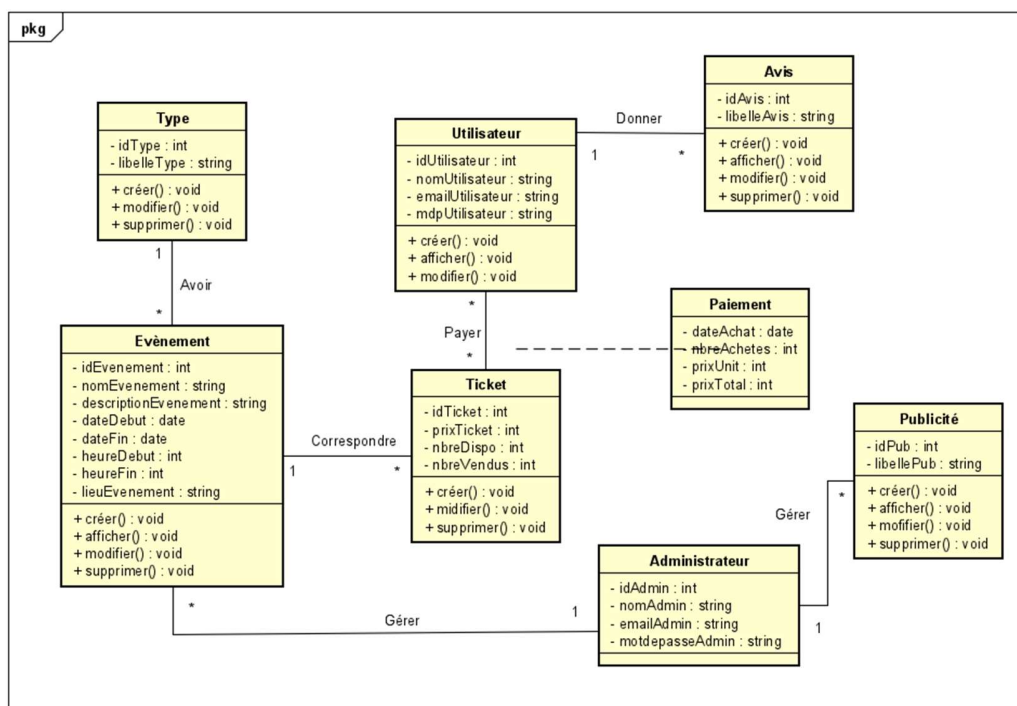


Figure 5 : Diagramme de classes

La classe Administrateur hérite de la classe Utilisateur, car l'administrateur est d'abord un utilisateur en plus de ses fonctions particulières. L'utilisateur peut créer plusieurs avis. Il

peut également acheter plusieurs tickets correspondant à plusieurs évènements différents. Un évènement a un type précis et plusieurs évènements peuvent être du même type. Un ticket correspond à un évènement précis. L'administrateur peut créer plusieurs publicités plusieurs newsletters.

III.3.2. Détails des classes

a. La classe Evènement

Elle est caractérisée par un identifiant, un nom, une description, un type, une période, une durée et un lieu. L'identifiant permet d'atteindre un évènement précis dans la base de données. La période est constituée de deux autres attributs : l'heure de début et l'heure de fin. La durée, elle, est constituée d'une date de début et d'une date de fin.

Les opérations réalisables sur cette table sont : la création, la modification, la suppression et l'affichage.

b. La classe Type

Elle regroupe les types associés aux évènements. Elle est caractérisée par un identifiant et un libellé. Les opérations réalisables sur cette table sont : la création, la modification, la suppression et l'affichage. Ces opérations sont réalisées uniquement par l'administrateur.

c. La classe Ticket

Elle est caractérisée par un identifiant, un prix, un nombre de tickets vendus et une quantité en stock. Les opérations réalisables sur cette table sont : la création, la modification et la suppression. Ces opérations sont réalisées uniquement par l'administrateur.

d. La classe Utilisateur

Elle est caractérisée par un identifiant, un nom, un email et un mot de passe. Les opérations réalisables sur cette table sont : la création, la modification, l'affichage et la suppression. Ces opérations sont réalisées uniquement par l'administrateur.

e. La classe Paiement

Elle est caractérisée par une date d'achat, un nombre de tickets achetés, un prix unitaire et le montant total de l'achat. Elle résulte de la relation n vers n entre les tables Utilisateur et Ticket. Elle s'instancie lorsque l'utilisateur effectue l'achat d'un ticket.

f. La classe Avis

Elle est caractérisée par un identifiant, un libellé et l'identifiant de l'utilisateur qui la crée.

g. La classe Administrateur

Elle est caractérisée par un identifiant, un nom, un email et un mot de passe. Aucune opération n'est réalisable sur cette table à travers le web service. C'est la classe qui peut manipuler la plupart des autres classes du système.

h. La classe Publicité

Elle est caractérisée par un identifiant, un libellé et l'identifiant de l'administrateur qui la crée.

III.4. Diagramme d'activités

Le diagramme d'activités est un diagramme comportemental d'UML, permettant de représenter le déclenchement d'évènements en fonction des états du système et de modéliser des comportements parallèles. Il se présente comme un organigramme qui décrit les enchaînements dans le déroulement d'un cas d'utilisation ou d'un ensemble de cas d'utilisation. Conceptuellement, il décrit un algorithme. Un diagramme d'activités permet de modéliser un processus interactif, global ou partiel pour un système donné (logiciel, système d'information). La description d'un cas d'utilisation par un diagramme d'activités correspond à sa traduction algorithmique. Une activité est l'exécution d'une partie du cas d'utilisation, elle est représentée par un rectangle aux bords arrondis. Le diagramme d'activités présente une vision macroscopique et temporelle du système modélisé.

III.4.1. Diagramme d'activités du cas : "S'authentifier"

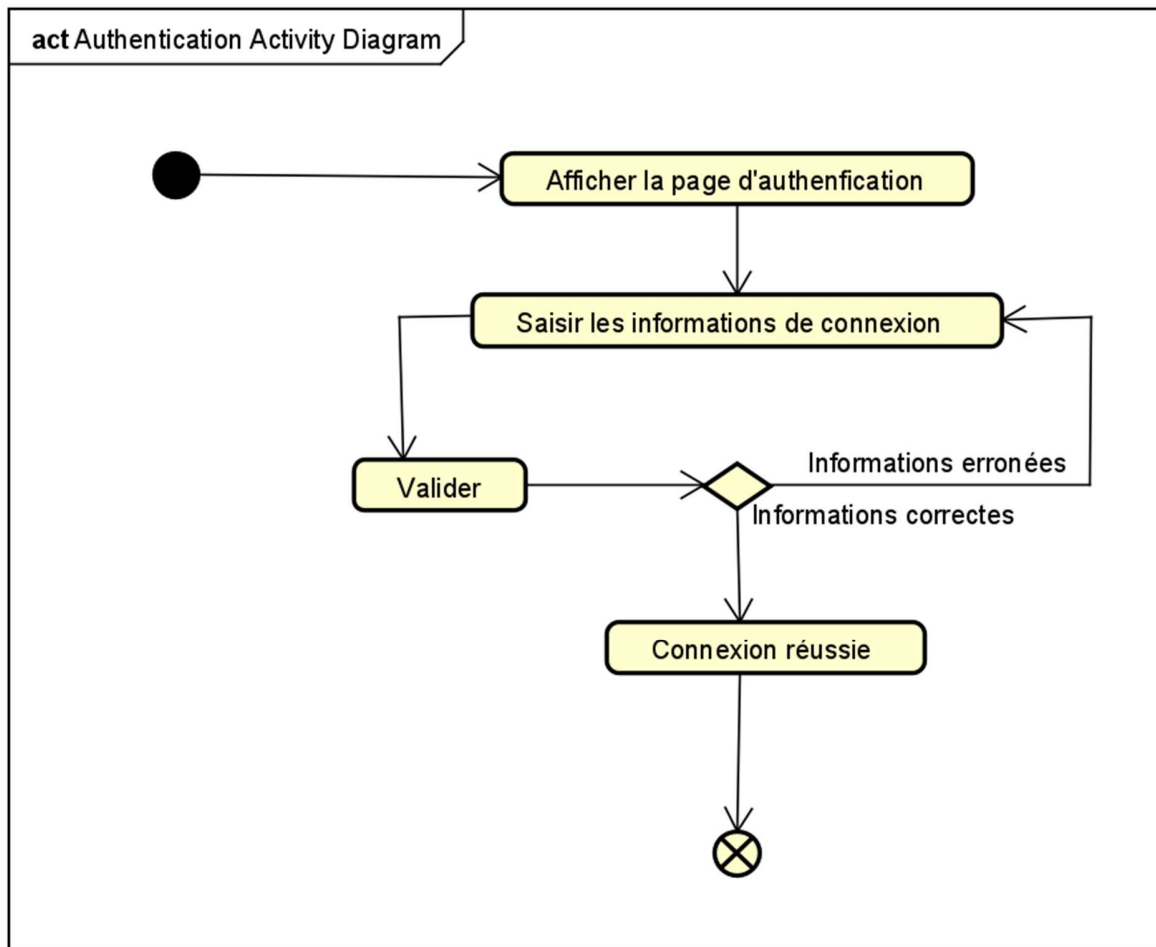


Figure 6 : Diagramme d'activités du cas : S'authentifier

Le cas d'utilisation « S'authentifier » démarre avec l'appel de la fonction afficher la page d'authentification par l'utilisateur. Lorsque la page lui est affichée, l'utilisateur saisit les informations de connexion et valide. Une vérification est faite par le système qui renvoie l'utilisateur à la saisie des informations en cas d'erreurs. Lorsque toutes ces fonctions se déroulent bien la connexion est réussie et le cas d'utilisation est terminé.

III.4.2. Diagramme d'activités du cas : "Créer un évènement"

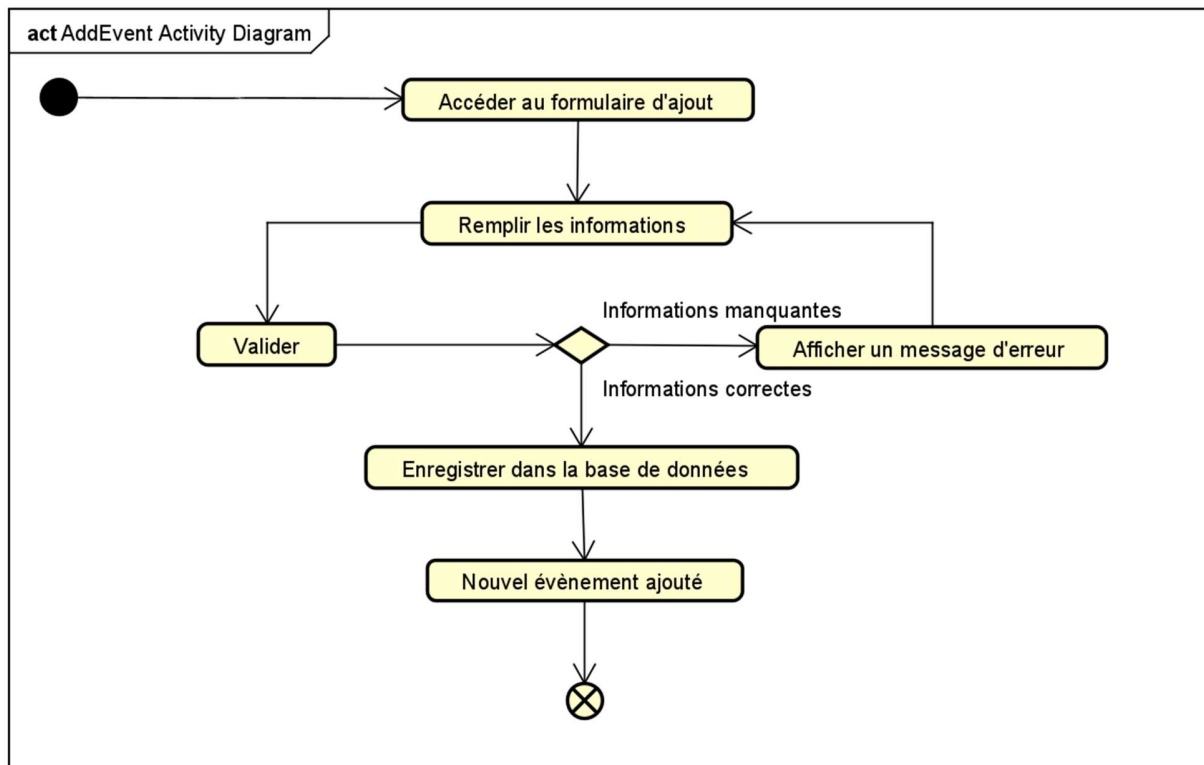


Figure 7 : Diagramme d'activités du cas : Ajouter un évènement

Pour réaliser le cas d'utilisation « Créer un évènement », l'administrateur affiche le formulaire de création par appel de la fonction afficher le formulaire à travers un bouton. Le système lui affiche donc la page et il remplit les informations concernant un évènement puis valide. Après une vérification effectuée par le système, un message d'erreur est affiché à l'administrateur en cas d'erreur puis il est redirigé vers la saisie des informations. Lorsque tout se déroule bien, les informations saisies sont envoyées et enregistrées dans la base de données et le cas d'utilisation prend fin.

III.4.3. Diagramme d'activités du cas : "Payer un ticket"

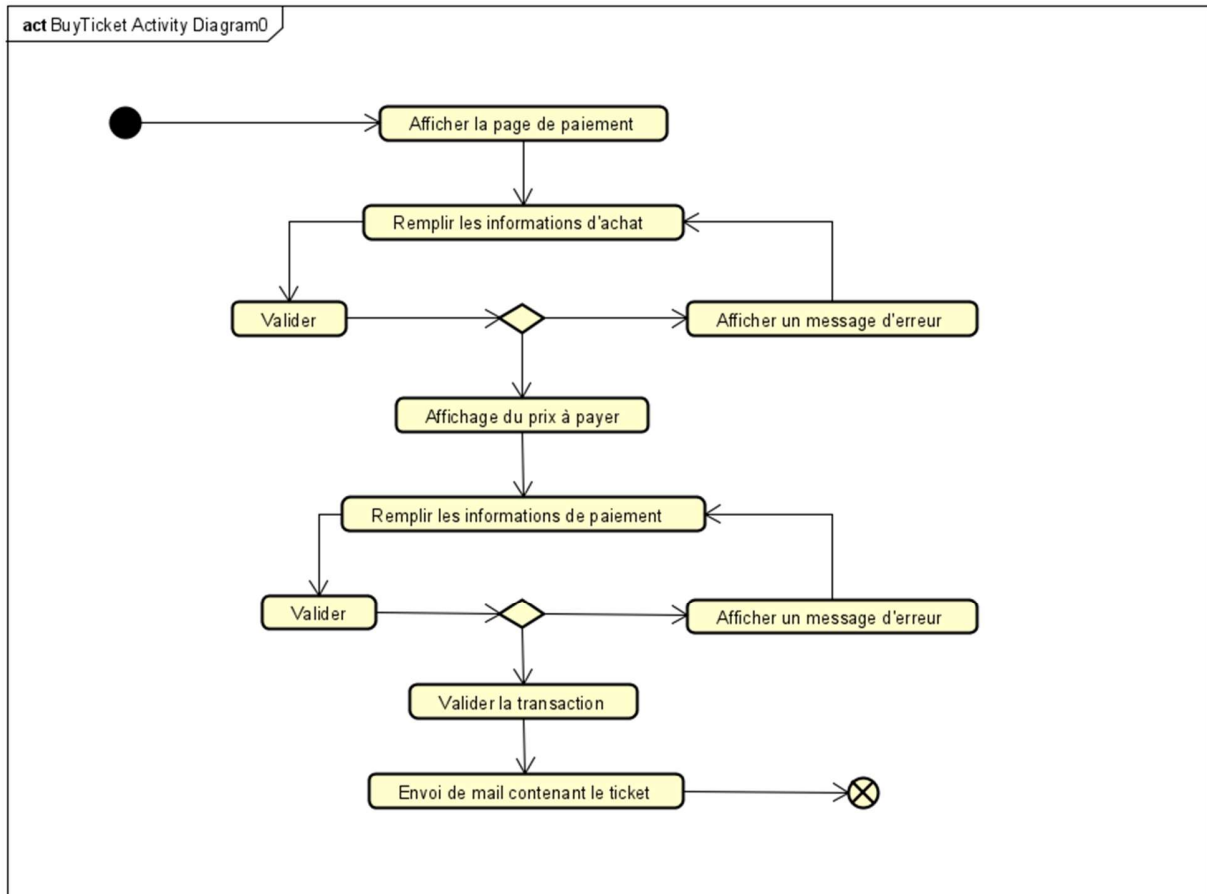


Figure 8 : Diagramme d'activités du cas : Payer un ticket

Le cas d'utilisation « Payer un ticket » démarre lorsque l'utilisateur déclenche la fonction afficher la page de paiement. Le système lui affiche alors une page d'achat où il renseigne les informations relatives à son achat puis valide. Après vérification par le système des informations, un message est affiché en cas d'erreur ; dans le cas contraire, le système calcule le montant total de l'achat et l'affiche. L'utilisateur remplit alors les informations de paiement telles que les coordonnées bancaires puis valide. Le système vérifie si le solde est suffisant ou pas. Dans le cas où le solde est insuffisant, un message est affiché à l'utilisateur et il rectifie sa saisie. Quand tout se déroule bien, l'utilisateur valide le paiement

et le système débite la carte puis lui envoie un mail contenant une facture et le ticket en question.

III.5. Diagramme de déploiement

Le diagramme de déploiement UML permet de montrer le déploiement physique des « artefacts » c'est-à-dire, des différentes composantes du système (fichiers, exécutables etc.), sur les ressources matérielles. A travers le diagramme de déploiement, le concepteur sait quels matériels il lui faut pour déployer l'application et surtout quelles parties du système vont sur tels matériels. Le diagramme de déploiement de notre solution est le suivant :

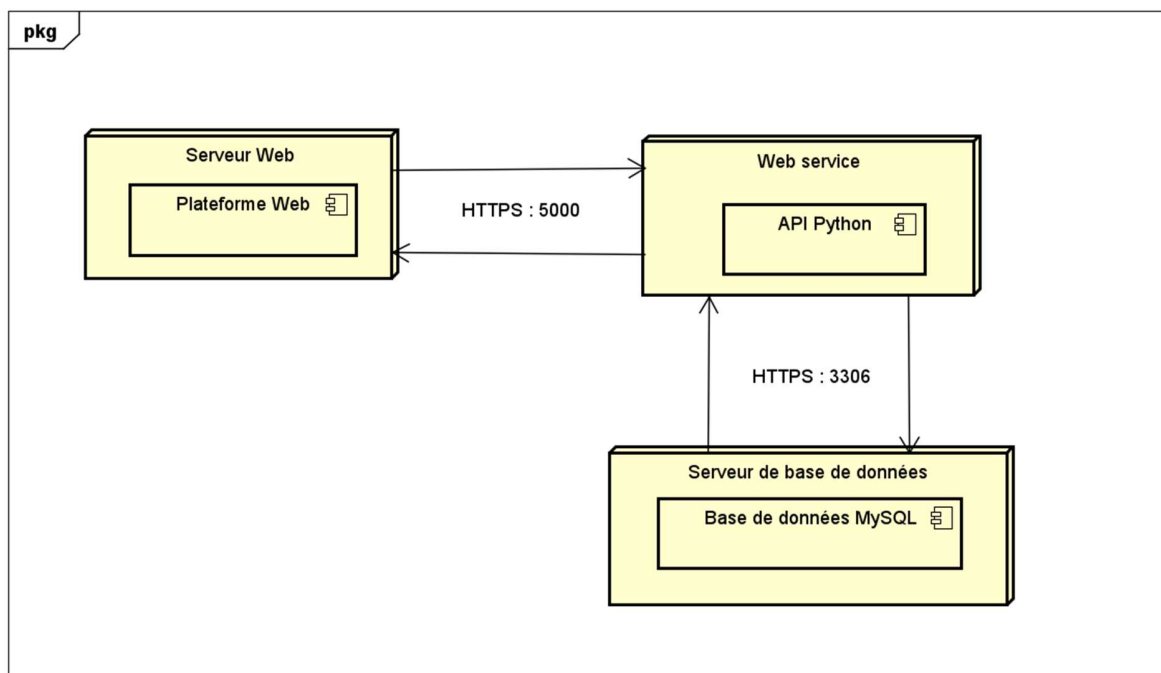


Figure 9 : Diagramme de déploiement

Notre système se compose de trois principaux blocs : le serveur Web, l'API et la base de données. Le serveur Web communique avec l'API en envoyant des requêtes en utilisant le protocole HTTPS sur le port 5000. L'API reçoit les demandes et consulte la base de données en utilisant le protocole HTTPS sur le port 3306. La base de données répond également en utilisant le même protocole sur le port 3306 également. L'API renvoie ensuite les résultats des requêtes vers le serveur Web en utilisant le protocole HTTPS sur le port 5000.